AVT Manta Cameras

AVT Manta Camera Controls

V4.1.0

29 June 2011

Allied Vision Technologies GmbH
Taschenweg 2a
D-07646 Stadtroda / Germany

ALLIED Vision Technologies

Legal notice

For customers in the U.S.A.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a residential environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However there is no guarantee that interferences will not occur in a particular installation. If the equipment does cause harmful interference to radio or television reception, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the distance between the equipment and the receiver.
- Use a different line outlet for the receiver.
- Consult a radio or TV technician for help.

You are cautioned that any changes or modifications not expressly approved in this manual could void your authority to operate this equipment. The shielded interface cable recommended in this manual must be used with this equipment in order to comply with the limits for a computing device pursuant to Subpart B of Part 15 of FCC Rules.

For customers in Canada

This apparatus complies with the Class B limits for radio noise emissions set out in the Radio Interference Regulations.

Pour utilisateurs au Canada

Cet appareil est conforme aux normes classe B pour bruits radioélectriques, spécifiées dans le Règlement sur le brouillage radioélectrique.

Life support applications

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Allied Vision Technologies customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Allied for any damages resulting from such improper use or sale.

Trademarks

Unless stated otherwise, all trademarks appearing in this document of Allied Vision Technologies are brands protected by law.

Warranty

The information provided by Allied Vision Technologies is supplied without any guarantees or warranty whatsoever, be it specific or implicit. Also excluded are all implicit warranties concerning the negotiability, the suitability for specific applications or the non-breaking of laws and patents. Even if we assume that the information supplied to us is accurate, errors and inaccuracy may still occur.

Copyright

All texts, pictures and graphics are protected by copyright and other laws protecting intellectual property. It is not permitted to copy or modify them for trade use or transfer, nor may they be used on web sites.

Headquarters:

Taschenweg 2A
D-07646 Stadtroda, Germany
Tel.: +49 (0)36428 6770
Fax: +49 (0)36428 677-28
e-mail: info@alliedvisiontec.com

Contents

Contacting Allied Vision Technologies

Info

- **Technical information:**
  http://www.alliedvisiontec.com
- **Support:**
  support@alliedvisiontec.com

Allied Vision Technologies GmbH (Headquarters)
Taschenweg 2a
07646 Stadtroda, Germany
Tel.: +49 36428-677-0
Fax.: +49 36428-677-28
e-mail: info@alliedvisiontec.com

Allied Vision Technologies Canada Inc.
101-3750 North Fraser Way
Burnaby, BC, V5J 5E9, Canada
Tel: +1 604-875-8855
Fax: +1 604-875-8856
e-mail: info@alliedvisiontec.com

Allied Vision Technologies Inc.
38 Washington Street
Newburyport, MA 01950, USA
Toll Free number +1 877-USA-1394
Tel.: +1 978-225-2030
Fax: +1 978-225-2029
e-mail: info@alliedvisiontec.com

Allied Vision Technologies Asia Pte. Ltd.
82 Playfair Road
#07-02 D'Lithium
Singapore 368001
Tel: +65 6634-9027
Fax: +65 6634-902
e-mail: info@alliedvisiontec.com

Introduction

The document **AVT Manta Camera Controls** describes the standard and advanced camera controls for AVT Manta cameras as seen from the **PvAPI GigE SampleViewer**.

Target group is the end user not the programmer.

The document **AVT Manta Camera Controls** answers questions about camera controls of the AVT GigE SDK and feature related items. Learn how to get more information at the AVT website, how to get information about software applicable with AVT Manta cameras and how to get deep information from the Manta Technical Manual.

Note

The AVT GigE SDK (and sample viewer) supports:

- Windows 7 (32 bit and 64 bit)
- Linux (32 bit and 64 bit)
- MacOS X
- QNX

Document history

Version	Date	Remarks
V2.0.3	15.04.10	NEW MANUAL - RELEASE status
V3.0.0	09.06.10	**Added Manta G-046, G-145, G-201, G-146:** • Table 35: Camera-specific exposure time offset on page 57 • Table 36: Camera-specific minimum exposure time on page 57 • Table 37: Jitter at exposure start (no binning, no sub-sampling) on page 58 • RGBA24 and BGRA24 are not supported, see Chapter PixelFormat on page 49. • ROI: RegionX/RegionY values only in steps of 2: see Table 27: Camera control: ROI on page 48

Table 1: Document history

Version	Date	Remarks
V4.0.0	25.10.10	**New file format:** • Changed file format from FM7 to FM9 **Added Manta G-033, G-504:** • Table 35: Camera-specific exposure time offset on page 57 • Table 36: Camera-specific minimum exposure time on page 57 • Table 37: Jitter at exposure start (no binning, no sub-sampling) on page 58
V4.0.1	05.01.11	**Some minor corrections** **Changed GigE Vision from V1.0 to V1.2:** • Chapter AVT software on page 11
V4.1.0	29.06.11	**Manta firmware upgrade:** • Added Mono12Packed and Bayer12Packed pixel formats: see Chapter PixelFormat on page 49. • Added Chapter Decimation (sub-sampling) on page 23ff. • Added Parameter FrameStartTriggerOverlap on page 38 • Added Mono12Packed on page 49 • Added Bayer12Packed on page 49 • Added Chapter Iris on page 66ff. (video auto-iris feature) • Added Chapter EventControl on page 80ff. • Added Chapter ChunkModeActive on page 84 • Added Chapter NonImagePayloadSize on page 84 • Added Chapter PayloadSize on page 84 • Added Chapter StreamFrameRateConstrain on page 85 • Added SyncInGlitchFilter on page 96 • Revised Chapter Gamma on page 56 • Added Chapter LUTControl on page 68

Table 1: Document history

Manual overview

The manual overview describes each chapter of this manual shortly.

- Chapter Contacting Allied Vision Technologies on page 6 lists AVT contact data for both:
  - Technical information / ordering
  - Commercial information
- Chapter Introduction on page 7 (this chapter) gives you the document history, a manual overview and conventions used in this manual (styles and

symbols). Furthermore you learn how to get more information on **AVT accessories**, available **AVT software** and the **AVT Manta Technical Manual**.

- Chapter AVT Manta camera controls on page 13 describes the features that are common for all AVT Manta cameras.
  - **Read this chapter before acquiring any images and working with any viewer.**
- Chapter Index on page 102 gives you quick access to all relevant data in this manual.

Conventions used in this manual

To give this manual an easily understood layout and to emphasize important information, the following typographical styles and symbols are used:

Styles

Style	Function	Example
Bold	Programs, inputs or highlighting important things	**bold**
Courier	Code listings etc.	`Input`
Upper case	Register	REGISTER
Italics	Modes, fields	*Mode*
Parentheses and/or blue	Links	(Link)

Table 2: Styles

Symbols

Note This symbol highlights important information.

Caution This symbol highlights important instructions. You have to follow these instructions to avoid malfunctions.

Caution-ESD

This symbol highlights important ESD instructions. Only **qualified personnel** is allowed to install and operate components marked with this symbol.

www

This symbol highlights URLs for further information. The URL itself is shown in blue.

Example:

http://www.alliedvisiontec.com

More information

In this chapter you get more information on **GigE Vision/GenICam**, **PvAPI**, **AVT accessories**, available **AVT software** and the **AVT Manta Technical Manual**.

GigE Vision and GenICam

- **Introduction to GigE Vision and GenICam**
  http://www.alliedvisiontec.com/emea/support/application-notes.html

PvAPI

- **PvAPI Manual: Prosilica PvAPI Programmers' Reference Manual**
  This manual is for programmers who want to work with the AVT Prosilica GigE SDK (PvAPI):
  http://www.alliedvisiontec.com/emea/products/software/windows/gige-sdk-pvapi.html

AVT Universal Package

- **AVT Universal Package User Guide**
  This manual is for programmers who want to work with the AVT GigE SDK (AVT UniAPI for all AVT 1394 and GigE Vision cameras):
  http://www.alliedvisiontec.com/emea/products/software/windows

AVT accessories

Note

Allied Vision Technologies offers a wide range of **accessories** for the use of AVT Manta cameras and the easy integration in already existing applications.

- **Gigabit Ethernet** accessories (Gigabit Ethernet network cards, hubs and switches, cables)
- **Lenses** (for cameras with sensors of type 1/3, 1/2, 2/3, 1, and 1.2)

www

For **more information on accessories** go to:

http://www.alliedvisiontec.com/emea/products/accessories.html

For **more information on lenses** go to:

http://www.alliedvisiontec.com/emea/products/accessories/lenses.html

To **order accessories online** (by clicking the article and sending an inquiry) visit the **AVT web shop** at:

http://www.alliedvisiontec.com/emea/products/accessories.html

AVT software

Note

AVT Manta cameras are **compliant to GigE Vision V1.2**. Moreover AVT Manta cameras offer many more functions than specified in the **GigE Vision V1.2** standard: so-called AVT smart features. You can use one of the following **AVT Software Packages**:

- **AVT PvAPI SDK** (see **AVT PvAPI Programmer's Reference Manual**)
- **AVT Universal Package (**see **AVT Universal Package User Guide**; not all smart features are supported)

All software packages provided by AVT are **free of charge** and contain the following components:

- Drivers
- Software Development Kit (SDK) for camera control and image acquisition
- Examples based on the provided APIs of the SDK
- Documentation and release notes
- Viewer application to operate/configure the cameras and access/test the **AVT smart features**.

www

All **software packages** (including **documentation** and **release notes**) provided by AVT can be downloaded at:

http://www.alliedvisiontec.com/emea/support/downloads/software.html

www

In addition to the AVT Software Packages Allied Vision Technologies offers special **Integration Packages** to integrate AVT cameras into any third-party vision software that supports the GigE Vision standard.

For more information refer to the **Software Package Selector Guide**.

Go to:

http://www.alliedvisiontec.com/emea/produkte/software.html

Here you also find the **AVT Software Packages for download** and **additional software documentation**:

- AVT Universal Package User Guide
- Release Notes

AVT Manta Technical Manuals

Note

Besides hardware installation procedures and the software documentation there is an in-depth description of all AVT Manta cameras in the **Technical Manual**:

- **MANTA Technical Manual**

Here you find: technical data, functional descriptions, features of the camera and how to use.

www

For **downloading the Technical Manuals** go to:

http://www.alliedvisiontec.com/emea/support/downloads/product-literature.html

AVT Manta camera controls

Note

Some features are not available for all camera models.

Example:

- White Balance is not available for monochrome cameras.

Some features are implemented in the cameras, but are not always available.

Example:

- Color correction features are implemented in MANTA cameras, but are not available in RAW mode.

Note

The camera controls are described as seen from the PvAPI (Prosilica SampleViewer).

Note

For a list of all implemented features (standard and advanced) see *Technical Manuals* of the AVT Manta cameras:

- **MANTA Technical Manual**, Chapter *Data path*, Table: *Available Manta camera controls*

http://www.alliedvisiontec.com/emea/support/downloads/product-literature.html

Note

For PvAPI users: attribute type is given in []:

- Enum
- Float32
- Uint32
- String
- Command

For more information see the **AVT PvAPI Programmers' Reference Manual**.

Note

The following abbreviations are used:

R/W = attribute is read/write

R/C = attribute is read only and constant

R/V = attribute is read only and volatile, can change at any time

Info

CameraName

Camera control	Parameter	Description
CameraName		[String] R/W The camera name can be modified by the user using the SampleViewer or *AVT Universal Package*. **Usage:** Use for multiple-camera situations for providing meaningful labels to individual cameras, e.g. *EngineRoomCam1*

Table 3: Camera control: **CameraName**

DeviceFirmwareVersion

Camera control	Parameter	Description
DeviceFirmwareVersion		[String] R/C Version of the firmware the camera is running.

Table 4: Camera control: **CameraName**

DeviceModelName

Camera control	Parameter	Description
DeviceModelName		[String] R/W Human readable model name, such as *Manta_G-125C*. Software should use the PartNumber and PartVersion to distinguish between models.

Table 5: Camera control: **DeviceModelName**

DevicePartNumber

Camera control	Parameter	Description
DevicePartNumber		[String] R/C Manufacturer's part number

Table 6: Camera control: **DevicePartNumber**

DeviceScanType

Camera control	Parameter	Description
DeviceScanType		[String] R/C Scan type of the camera, e.g. *Areascan*.

Table 7: Camera control: **DeviceScanType**

DeviceSerialNumber

Camera control	Parameter	Description
DeviceSerialNumber		[String] R/C Serial number of the camera. *DeviceSerialNumber* is not a unique identifier across models; software should use *UniqueId* instead.

Table 8: Camera control: **DeviceSerialNumber**

DeviceVendorName

Camera control	Parameter	Description
DeviceVendorName		[String] R/C Manufacturer's name: Allied Vision Technologies

Table 9: Camera control: **DeviceVendorName**

Firmware

These camera controls are read only.

Relates to information about the firmware that is currently on the camera.

Camera control	Parameter	Description
Firmware	FirmwareVerBuild	[Uint32] R/C Build information
	FirmwareVerMinor	[Uint32] R/C The minor part of firmware version number (part after the decimal)
	FirmwareVerMajor	[Uint32] R/C The major part of the firmware version number (part before the decimal)

Table 10: Camera control: **Firmware**

Part

When communicating with AVT about an AVT Manta camera, note first the serial number.

Camera control	Parameter	Description
PartClass		[Uint32] R/C Camera part class (manufacturer dependant).
PartNumber		[Uint32] R/C Camera part number. Manufacturer part number for the camera model.
PartRevision		[String] R/C Camera revision. Part number revision level.

Table 11: Camera control: **Part**

Camera control	Parameter	Description
PartVersion		[String] R/C Camera version. Part number version level.
SerialNumber		[String] R/C Camera serial number. You can read out this number with the read-only *DeviceID* camera control.

Table 11: Camera control: **Part**

Sensor

Information about the image sensor.

Camera control	Parameter	Description
Sensor	SensorBits	[Uint32] R/C The sensor digitization bit depth.
	SensorHeight	[Uint32] R/C The total number of pixel rows on the sensor.
	SensorType	[Enum] R/C Monochrome or Bayer-pattern color sensor type.
	SensorWidth	[Enum] R/C The total number of pixel columns on the sensor.

Table 12: Camera control: **Sensor**

UniqueID

Camera control	Parameter	Description
UniqueID		[Uint32] R/C The unique camera ID that differentiates the current camera from all other cameras.

Table 13: Camera control: **UniqueID**

ImageMode

This camera control provides the binning and the decimation (sub-sampling) features.

Binning

Definition **Binning** is the summing of charge of adjacent pixels on a sensor, to give a lower resolution but more sensitive image. AVT's Manta CCD cameras have independent x,y binning. The 2x/4x/8x binning which are described on the following pages are only examples. Indeed: AVT's Manta cameras have 1x, 2x, 3x, 4x, 5x, ... binning in each direction (x and y).

BinningX up to 8x.
BinningY up to 14x.

Where done? BinningX (horizontal binning) is done in the FPGA.
BinningY (vertical binning) is done directly in the sensor.

Camera control	Parameter	Description
ImageMode	BinningX	[Uint32] R/W BinningX sets the horizontal binning. Binning is very useful for dramatically increasing the camera sensitivity and frame rate at the expense of resolution.
	BinningY	[Uint32] R/W BinningY controls the vertical binning. Vertical binning is useful for cases where the horizontal resolution is important, but where high-sensitivity can be achieved by using vertical binning. In most cases, BinningX and BinningY would be set to equal values.

Table 14: Camera control: **ImageMode**

Useful Use binning primarily for 3 reasons:

- A reduction in the number of pixels and thus the amount of data while retaining the original image area angle
- An increase in the frame rate (BinningY only)

- A brighter image, also resulting in an improvement in the signal-to-noise ratio of the image

Signal-to-noise ratio (SNR) and **signal-to-noise separation** specify the quality of a signal with regard to its reproduction of intensities. The value signifies how high the ratio of noise is in regard to the maximum achievable signal intensity.

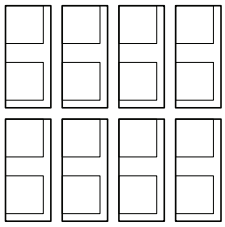
The higher this value, the better the signal quality. The unit of measurement used is generally known as the decibel (dB), a logarithmic power level. 6 dB is the signal level at approximately a factor of 2.

However, the advantages of increasing signal quality are accompanied by a reduction in resolution.

BinningY

BinningY increases the light sensitivity of the camera by a factor of two by adding together the values of two adjoining vertical pixels output as a single pixel. This is done directly in the horizontal shift register of the sensor.

This reduces vertical resolution, depending on the model.



BinningY factor 2

Figure 1: 2 x vertical binning

Note

Vertical resolution is reduced, but **signal-to-noise ratio** (SNR) is increased by about 3, 6 or 9 dB (2 x, 4 x or 8 x binning).

Note

The image appears **vertically** compressed in this mode and no longer exhibits a true aspect ratio.

If **vertical binning** is activated the image may appear to be over-exposed and may require correction.

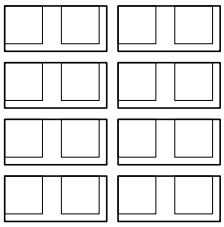
Caution

Although binning is possible with color cameras, it is not recommended, because the color information will be destroyed.

BinningX

BinningX increases the light sensitivity of the camera by adding together the values of two adjoining horizontal pixels output as a single pixel. This is done digitally in the FPGA.

This reduces horizontal resolution, depending on the model. With 2x BinningX the light sensitivity is increased by a factor of two (6 dB). Signal-to-noise separation improves by approx. 3 dB.



BinningX factor 2

Figure 2: 2 x horizontal binning

Note

The image appears **horizontally** compressed in this mode and does no longer show true aspect ratio.

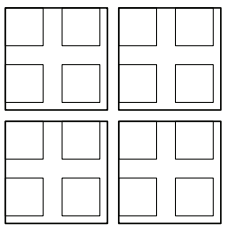
If **horizontal binning** is activated the image may appear to be over-exposed and may require correction.

BinningY and BinningX combined (full binning)

If horizontal and vertical binning are combined, every 4 pixels are consolidated into a single pixel. At first two vertical pixels are put together and then combined horizontally.

This increases light sensitivity by a total of a factor of 4 (16 or 64) and at the same time signal-to-noise separation is improved by about 6 (12 or 18) dB.

Resolution is reduced, depending on the model.



BinningY and BinningX: factor 2

Figure 3: 2 x full binning

Decimation (sub-sampling)

Decimation (also known as sub-sampling) is available for Manta b/w and color cameras. (Firmware 1.44 or greater)

Note **Manta G-032 cameras don't have decimation.**

What is decimation?

Definition Decimation (also known as sub-sampling) is the process of skipping neighboring pixels (with the same color) while being read out from the CCD chip.

Which Manta models have decimation?

All Manta models (except Manta G-032), both color and b/w, have this feature.

Camera control	Parameter	Description	
ImageMode	DecimationHorizontal	Horizontal sub-sampling of the image. This reduces the **horizontal** resolution (width) of the image by the specified horizontal decimation factor. No increased frame rate.	[Integer] 1 = off 2 = 2 out of 4 decimation 3/4 = 2 out of 8 decimation 5/6/7/8 = 2 out of 16 decimation Each combination of vertical binning and horizontal decimation or vice versa is possible. But: • Combination of horizontal binning + horizontal decimation is not possible. • Combination of vertical binning + vertical decimation is not possible.
	DecimationVertical	Vertical sub-sampling of the image. This reduces the **vertical** resolution (width) of the image by the specified vertical decimation factor. Increased frame rate.	

Table 15: Camera control: **ImageMode**

Description of decimation

Decimation is used primarily for the following reason:

- A reduction in the number of pixels and thus the amount of data while retaining the original image area angle and image brightness

Similar to binning mode the cameras support horizontal, vertical and H+V decimation mode.

Horizontal decimation modes

The different decimation patterns are shown below.

2 out of 4

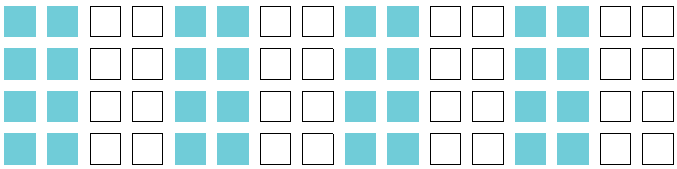


Figure 4: Horizontal decimation 2 out of 4 (**b/w**)

2 out of 8

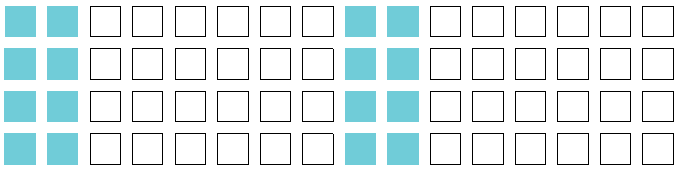


Figure 5: Horizontal decimation 2 out of 8 (**b/w**)

2 out of 16

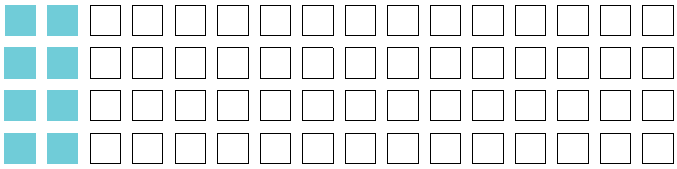


Figure 6: Horizontal decimation 2 out of 16 (**b/w**)

2 out of 4

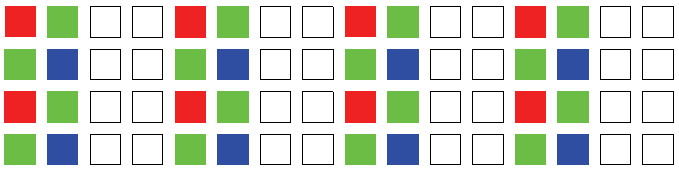


Figure 7: Horizontal decimation 2 out of 4 (**color**)

2 out of 8

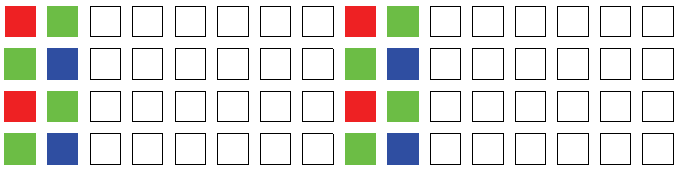


Figure 8: Horizontal decimation 2 out of 8 (**color**)

2 out of 16

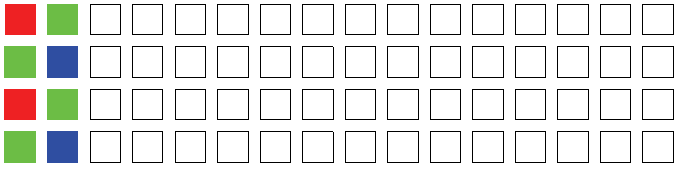


Figure 9: Horizontal decimation 2 out of 16 (**color**)

Note

The image appears horizontally compressed in this mode and no longer exhibits a true aspect ratio.

Vertical decimation modes

The different decimation patterns are shown below.

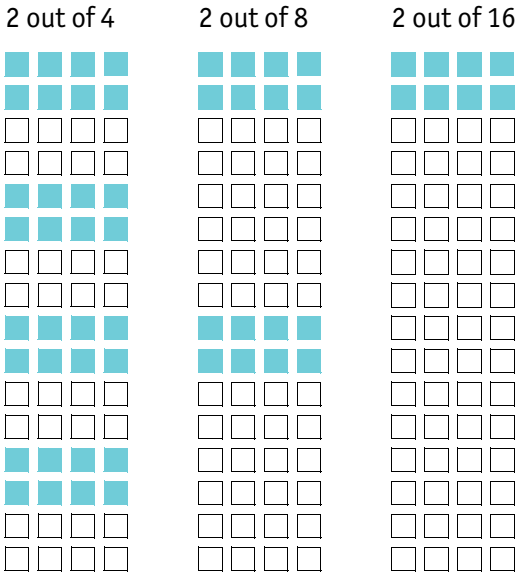


Figure 10: Vertical decimation (**b/w**)

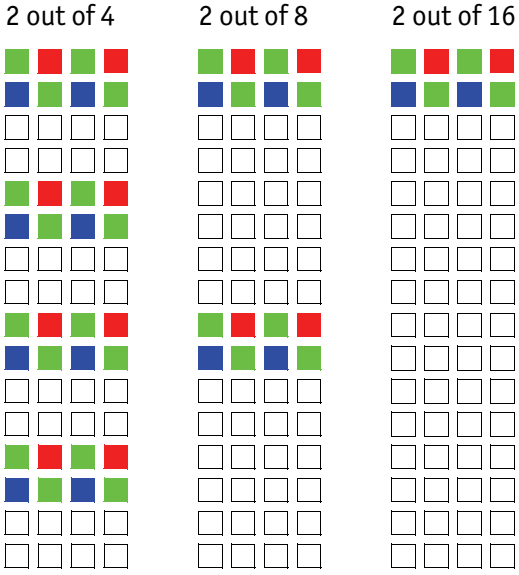


Figure 11: Vertical decimation (**color**)

Note **The image appears vertically compressed in this mode and no longer exhibits a true aspect ratio.**

H+V decimation modes

The different decimation patterns are shown below.

2 out of 4 H+V decimation

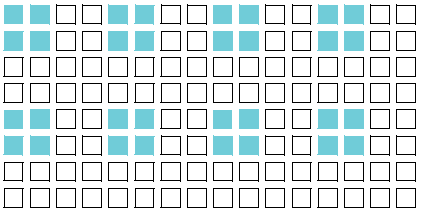


Figure 12: 2 out of 4 H+V decimation (**b/w**)

2 out of 8 H+V decimation

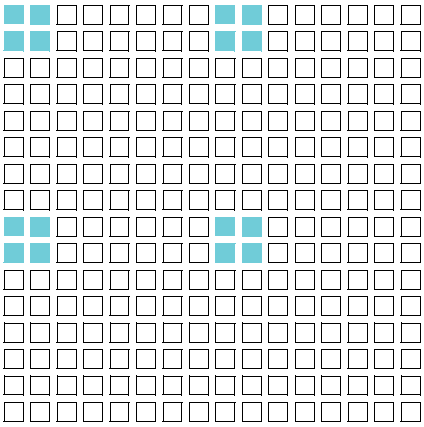


Figure 13: 2 out of 8 H+V decimation (**b/w**)

2 out of 16 H+V decimation

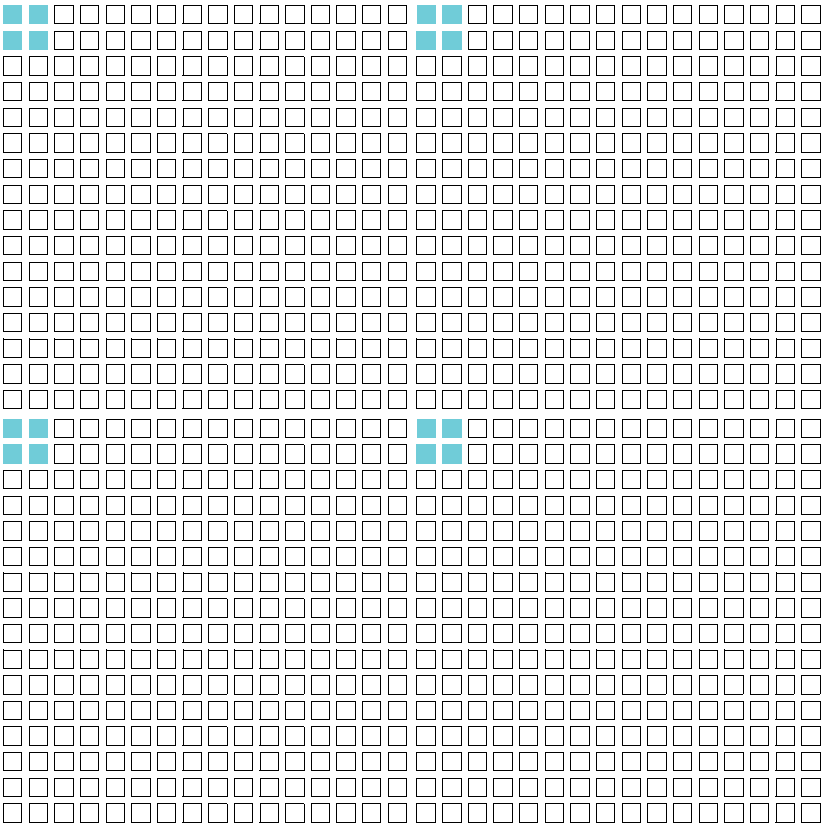


Figure 14: 2 out of 16 H+V decimation (**b/w**)

2 out of 4 H+V decimation

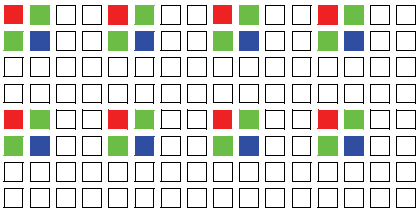


Figure 15: 2 out of 4 H+V decimation (**color**)

2 out of 8 H+V decimation

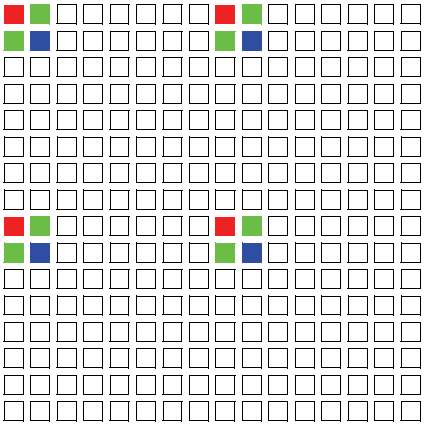


Figure 16: 2 out of 8 H+V decimation (**color**)

2 out of 16 H+V decimation

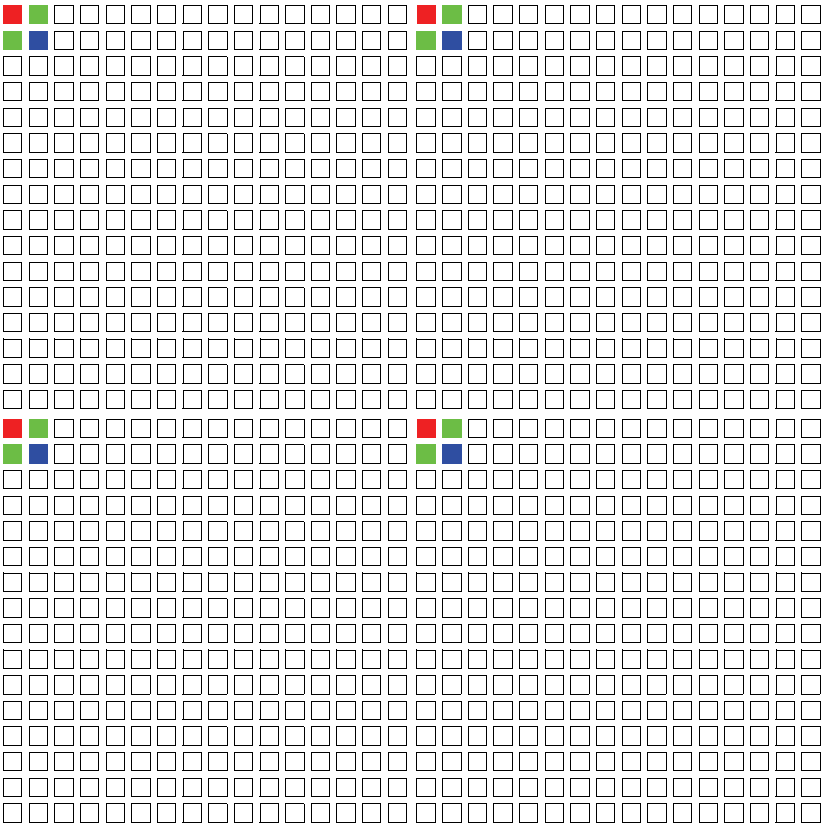


Figure 17: 2 out of 16 H+V decimation (**color**)

Acquisition

Note

For a description how to use the trigger related camera controls in combination and common problems while using the triggers see Chapter Trigger concept for advanced users on page 43.

This group of controls relates to the image acquisition.

Trigger

This group of controls relates to how an image frame is initiated or triggered.

AcqEnd

Camera control	Parameter	Description
AcqEnd	AcqEndTriggerEvent	[Enum] R/W If AcqEndTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger. Defines what type of external input trigger will end acquisition.
		EdgeRising Rising edge trigger
		EdgeFalling Falling edge trigger
		EdgeAny Rising or falling edge
		LevelHigh Active high signal
		LevelLow Active low signal

Table 16: Camera control: **AcqEnd**

Camera control	Parameter	Description
AcqEnd	AcqEndTriggerMode	[Enum] R/W Selects if the end of acquisition should be stimulated by an external hardware trigger. See the *AcquisitionStop* command for software triggering.
		SyncIn1 Trigger at *SyncIn1* to be associated with this control.
		SyncIn2 Trigger at *SyncIn2* to be associated with this control.
		Disabled No external trigger. Acquisition must be stopped with the AcquisitionStop API command.

Table 16: Camera control: **AcqEnd**

AcqRec

An AcqStart hardware trigger signal, or the AcquisitionStart command, must be received before your AcqRec trigger. See AcquisitionMode = Recorder.

Camera control	Parameter	Description
AcqRec	AcqRecTriggerEvent	[Enum] R/W What kind of external input trigger will start a recording sequence when *AcquisitionMode* set to *Recorder*.
		EdgeRising Rising edge trigger
		EdgeFalling Falling edge trigger
		EdgeAny Rising or falling edge
		LevelHigh Active high signal
		LevelLow Active low signal
AcqRec	AcqRecTriggerMode	[Enum] R/W Selects if the start of a Recorder event should be stimulated by an external hardware trigger. There is no software trigger event capability for this mode.
		SyncIn1 Trigger at *SyncIn1* to be associated with this control
		SyncIn2 Trigger at *SyncIn2* to be associated with this control
		Disabled No external trigger. Unlike AcqStart and AcqEnd, there is no API command trigger option for a recording event.

Table 17: Camera control: **AcqRec**

AcqStart

Controls relating to the start of an acquisition stream. Frames are triggered within this acquisition stream (see FrameStart).

Camera control	Parameter	Description
AcqStart	AcqStartTriggerEvent	[Enum] R/W If AcqStartTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.
		EdgeRising Rising edge trigger
		EdgeFalling Falling edge trigger
		EdgeAny Rising or falling edge
		LevelHigh Active high signal
		LevelLow Active low signal
AcqStart	AcqStartTriggerMode	[Enum] R/W Selects if the start of acquisition should be stimulated by an external hardware trigger. See the AcquisitionStart command for software triggering.
		SyncIn1 Trigger at *SyncIn1* to be associated with this control
		SyncIn2 Trigger at *SyncIn2* to be associated with this control
		Disabled No external trigger Acquisition must be started with the AcquisitionStart API command.

Table 18: Camera control: **AcqStart**

FrameRate

Camera control	Parameter	Description
FrameRate		[Float32] R/W When *FrameStartTriggerMode* is set to *FixedRate*, this control specifies the frame rate.

Table 19: Camera control: **FrameRate**

FrameStart

Various methods and controls for starting image capture can be configured here. Controls are described relating to the triggering of frames within an acquisition stream.

Camera control	Parameter	Description
FrameStart	FrameStartTriggerDelay	[Uint32] R/W Start-of-image can be delayed [in microseconds] to begin some time after a trigger event is received by the camera. This feature is valid only when *FrameStartTriggerMode* is set to external trigger (i.e. *SyncIn1*, *SyncIn2*). Normally, this delay value is set to zero. **Usage:** This control is a common trigger to sync with a strobe lighting source, which will inherently have some fixed setup time.
	FrameStartTriggerEvent	[Enum] R/W If FrameStartTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.
		EdgeRising Rising edge trigger
		EdgeFalling Falling edge trigger
		EdgeAny Rising or falling edge
		LevelHigh Active high signal
		LevelLow Active low signal

Table 20: Camera control: **FrameStart**

Camera control	Parameter	Description
FrameStart	FrameStartTriggerMode	[Enum] R/W Determines how an image frame is initiated within an acquisition stream. **Note** For *Freerun* and *FixedRate* the first frame is synchronized to the AcqStart event.
		Freerun Camera runs at maximum supported frame rate depending on the exposure time and region of interest (ROI) size. **Note** *StreamBytesPerSecond* can slow down the maximum supported frame rate.
		SyncIn1 External trigger SyncIn1
		SyncIn2 External trigger SyncIn2
		FixedRate Camera self-triggers at a fixed frame rate defined by FrameRate.
		Software Software initiated image capture

Table 20: Camera control: **FrameStart**

Camera control	Parameter	Description
FrameStart	FrameStartTriggerOverlap	[Enum] R/W Possible values: *Off*, *PreviousFrame* • When *Off*, any external trigger received before *FrameTriggerReady* signal is high is ignored. • When *PreviousFrame*, any external trigger received before *FrameTriggerReady* is latched and used to trigger the next frame. Default is *Off*.
	FrameStartTriggerSoftware	[Command] Triggers a frame within acquisition stream. Valid when *FrameStartTriggerMode* = *Software*

Table 20: Camera control: **FrameStart**

AcquisitionAbort

Camera control	Parameter	Description
AcquisitionAbort		[Command] **Usage:** Use to abort the current acquisition.

Table 21: Camera control: **AcquisitionAbort**

AcquisitionFrameCount

Camera control	Parameter	Description
AcquisitionFrameCount		[Uint32] R/W Use to define the number of frames to capture when capturing a limited sequence of images. Use in combination with *MultiFrame* and *Recorder acquisition* modes.

Table 22: Camera control: **AcquisitionFrameCount**

AcquisitionMode

The acquisition modes determine how the camera handles frame triggers within the acquisition stream.

Camera control	Parameter	Description
AcquisitionMode		[Enum] R/W
	Continuous	This is the normal acquisition mode of the camera. After an acquisition start event, the camera will continuously receive frame trigger events, or in the case where *FrameStartTriggerMode* equals *Freerun*, will continuously stream.
	SingleFrame	In this mode, the camera will only deliver a single image. So in triggered mode, a single frame will be delivered on the first trigger event, but no further frames will be delivered on subsequent trigger events (until acquisition is stopped and restarted).

Table 23: Camera control: **AcqisitionMode**

Camera control	Parameter	Description
AcquisitionMode	MultiFrame	In this mode, a single trigger event will cause the camera to deliver a specific number of frames based on *AcquisitionFrameCount*. The camera frame rate is defined by *FrameStart* settings. Further trigger events will be ignored until acquisition is stopped and restarted.
	Recorder	In this mode, the camera will continuously record images into the camera on-board memory but will not send them to the host until an *AcqRec* trigger signal is received. Further *AcqRec* trigger events will be ignored until acquisition is stopped and restarted. **Usage:** Combined with the *RecorderPreEventCount* control, this feature is useful for returning any number of frames **before** a trigger event. When *AcqRec* trigger is received, the currently imaging/acquiring image will complete as normal, and then at least one more image will be taken. See Chapter RecorderPreEventCount on page 42. The memory is a circular buffer, that starts rewriting images once it is full. Its size is determined by *AcquisitionFrameCount*.

Table 23: Camera control: **AcqisitionMode**

AcquisitionStart

Camera control	Parameter	Description
AcquisitionStart		[Command] **Usage:** starts the camera imaging.

Table 24: Camera control: **AcquisitionStart**

AcquisitionStop

Camera control	Parameter	Description
AcquisitionStop		[Command] **Usage:** stops the image stream.

Table 25: Camera control: **AcquisitionStop**

RecorderPreEventCount

Camera control	Parameter	Description
RecorderPreEventCount		[Uint32] R/W The number of frames that should proceed the trigger event when using the camera as an event recorder. Manta cameras have a post-event trigger capability whereby frames from before a trigger event can be captured. Use together with Recorder *AcquisitionMode* and *AcquisitionFrameCount*. Valid only when *AcquisitionMode* = *Recorder*. **Note** At least one image must be captured after the *AcqRec* trigger event. That is, you **cannot** set RecorderPreEventCount=1, AcquisitionFrameCount=1.

Table 26: Camera control: **RecorderPreEventCount**

Trigger concept for advanced users

Note

For an introduction into the general trigger concept and trigger definitions see **Manta Technical Manual**, chapters *Trigger timing diagram* and chapter *Notes on triggering*.

The acquisition/frame concept

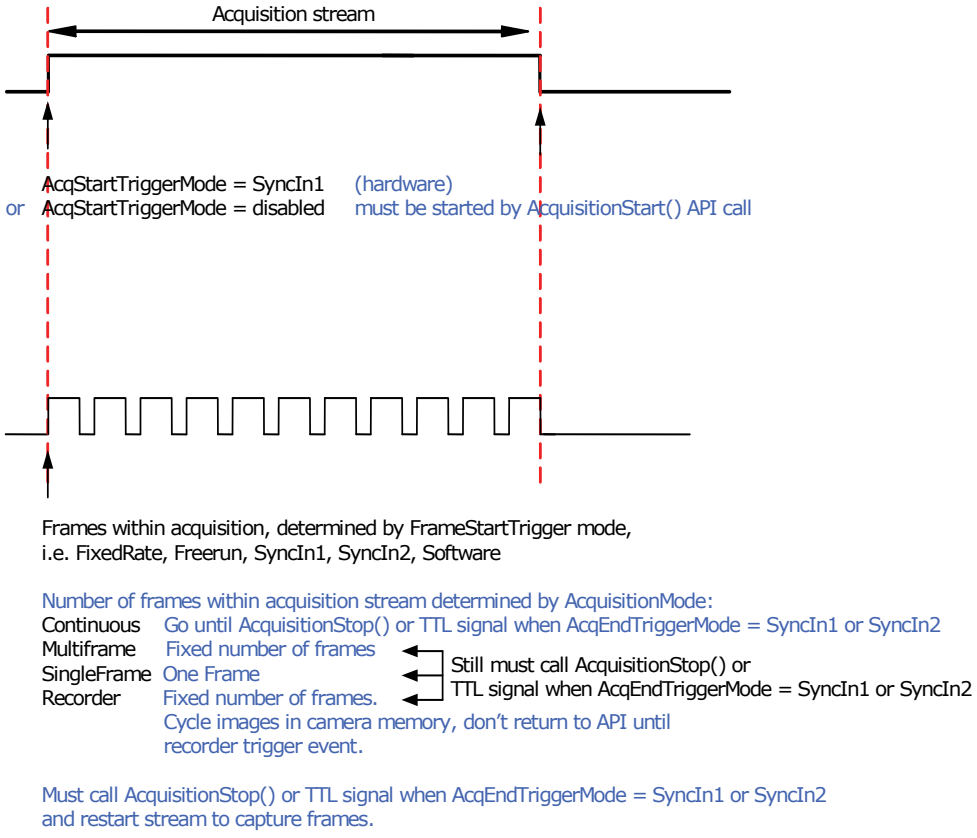


Figure 18: AcquisitionStart and AcquisitionAbort

Scenario 1: Acquisition controlled by hardware trigger (Freerun)

AcqStartTriggerMode = SyncIn1
FrameStartTriggerMode = Freerun
AcquisitionMode = Freerun
AcqEndTriggerMode = SyncIn1

AcqStartTriggerEvent = EdgeRising or LevelHigh

AcqEndTriggerEvent = EdgeFalling or LevelLow

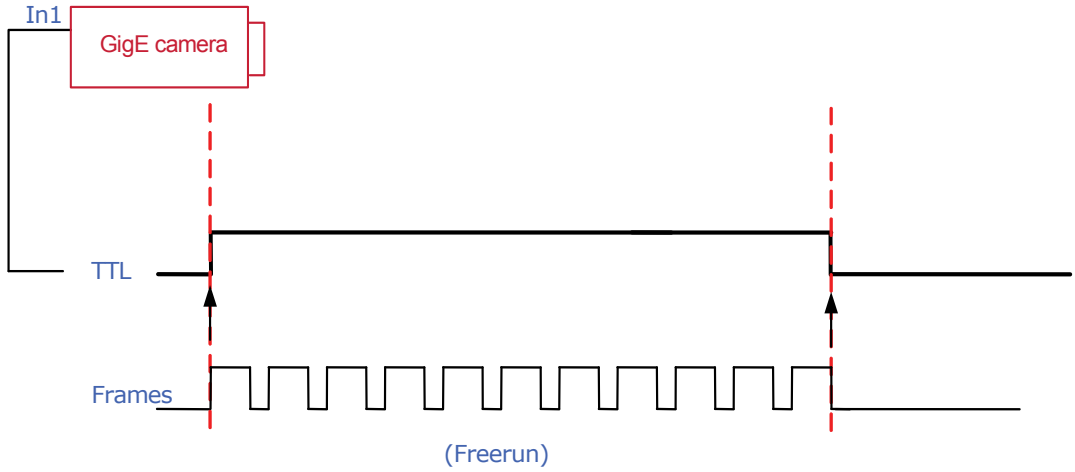


Figure 19: Trigger scenario 1: Acquisition controlled by hardware trigger (Freerun)

Scenario 2: Controlling exposure duration by external trigger

AcqStartTriggerMode/AcqEndTriggerMode = Disabled
FrameStartTriggerMode = SyncIn1
AcquisitionMode = MultiFrame
ExposureMode = External
AcquistionFrameCount = 3

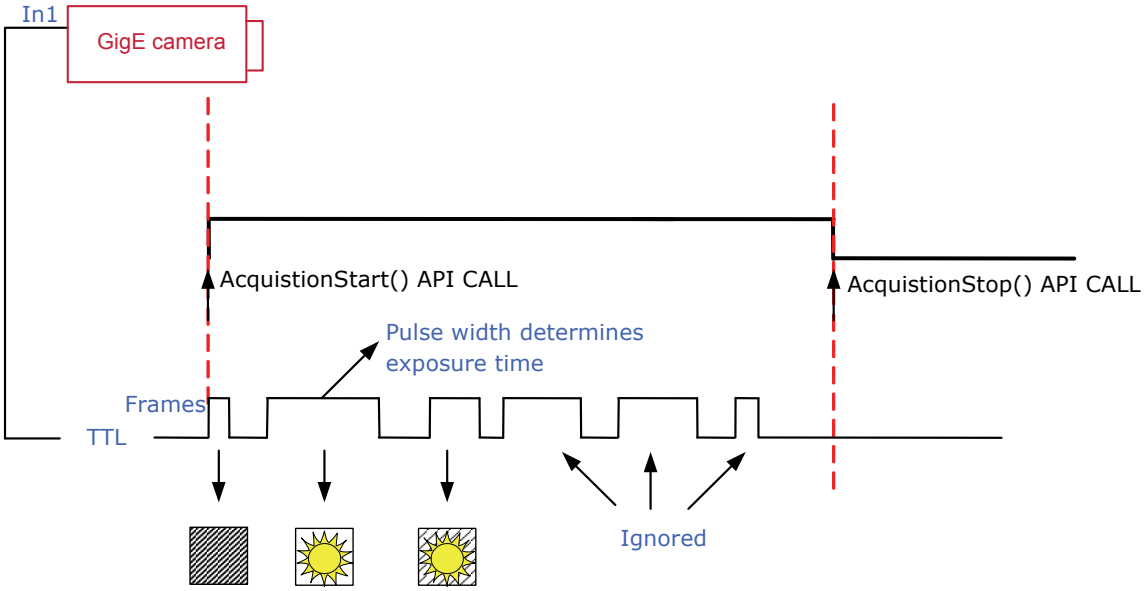


Figure 20: Trigger scenario 2: Controlling exposure duration by external trigger

Scenario 3: Recorder mode

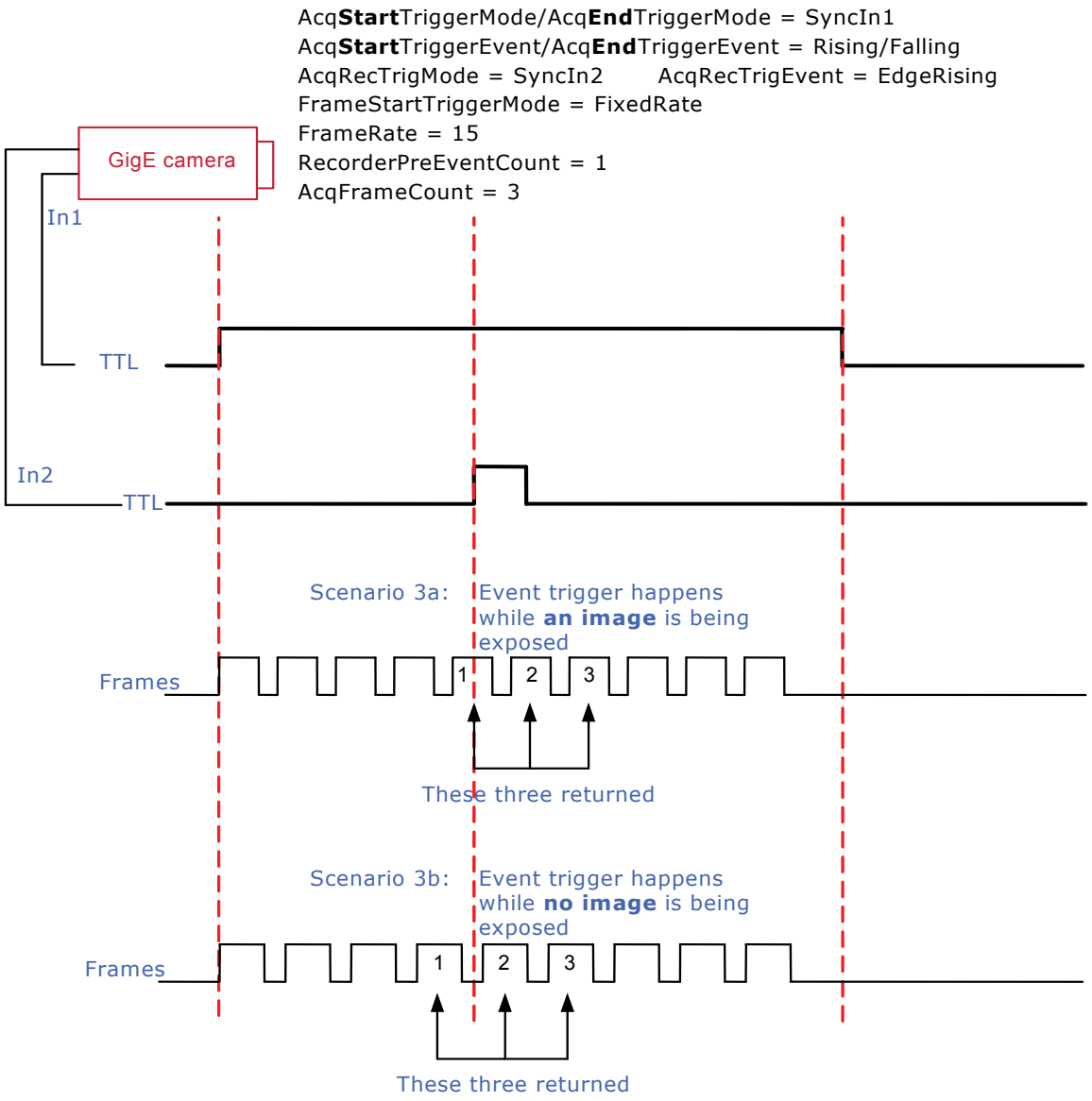


Figure 21: Trigger scenario 3: Recorder mode

Which command controls frames?

Note

You have to distinguish between **AcqStart** and **FrameStart**.

Only FrameStartTriggerMode controls frames.

See the following description.

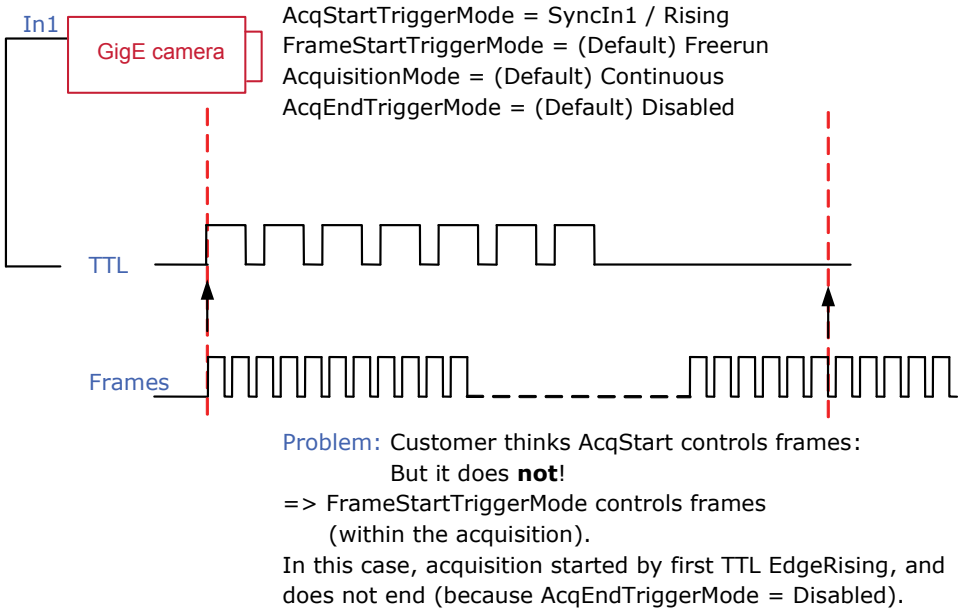


Figure 22: Which command controls frames?

ImageFormat

ROI

Definition Region of Interest (**ROI**) defines a rectangular sub-region of the image. Selecting an ROI that is small can increase the maximum frame rate and reduce the amount of image data. The following parameters define the size and location of the ROI sub-region:

Camera control	Parameter	Description
ROI	Height	[Uint32] R/W Specifies, in rows, the vertical size of the rectangle that defines the ROI.
	RegionX	[Uint32] R/W Specifies, in pixels, the position of the top-left corner of the ROI (0,0) In steps of 2 only, beginning with 0. If an odd value is entered, then this value is rounded down to the next even value.
	RegionY	[Uint32] R/W Specifies, in pixels, the position of the top-left corner of the ROI (0,0) In steps of 2 only, beginning with 0. If an odd value is entered, then this value is rounded down to the next even value.
	Width	[Uint32] R/W Defines, in columns, the horizontal size of the rectangle that defines the ROI.

Table 27: Camera control: **ROI**

PixelFormat

There are various pixel data formats that AVT Manta cameras can output. Not all cameras have every mode (see the **Technical Manuals**, Chapter *Specification* for details):

Camera control	Parameter	Description
PixelFormat		[Enum] R/W
	Mono8	8 bits per pixel, monochrome. On-camera interpolation, with luminance (Y) channel returned.
	Mono12Packed	12 bits per pixel. 2 pixels of data every 3 bytes. Monochrome.
	Mono16	16 bits per pixel, monochrome. On-camera interpolation, with luminance (Y) channel returned. Data is least significant bit aligned within a 16bit unsigned integer, e.g. for 12 bit camera: 0000xxxx xxxxxxxx.
	Bayer8	8 bits per pixel, un-interpolated color
	Bayer12Packed	12 bits per pixel. 2 pixels of data every 3 bytes. Raw un-interpolated data from camera.
	Bayer16	16 bits per pixel, un-interpolated color. Data is least significant bit aligned within a 16-bit unsigned integer, e.g. for 12-bit camera: 0000xxxx xxxxxxxx
	RGB24	24 bits per pixel, on-camera interpolated color Data sent as three consecutive bytes, representing B, G, R of a pixel.
	BGR24	24 bits per pixel, on-camera interpolated color Data sent as three consecutive bytes, representing B, G, R of a pixel.
	YUV411	Data sent in YUV format. On-camera interpolated color. (12 bits per pixel)
	YUV422	Data sent in YUV format. On-camera interpolated color. (16 bits per pixel)
	YUV444	Data sent in YUV format. On-camera interpolated color. (24 bits per pixel)

Table 28: Camera control: **PixelFormat**

TotalBytesPerFrame

Camera control	Parameter	Description
TotalBytesPerFrame		[Uint32] R/V The total number of bytes per image frame. Dependant on ROI, PixelFormat, and Binning.

Table 29: Camera control: **TotalBytesPerFrame**

Controls

ColorTransformationControl (only color models)

Note

Some features are implemented in the cameras, but are not always available.

Example:

- Color correction features are implemented in MANTA cameras, but are not available in RAW mode.

The ColorTransformationControl section describes features related to color transformations in the AVT GigE cameras.

Definition

The **color transformation** is a linear operation taking as input the triplet R_{in}, G_{in}, B_{in} for an RGB color pixel. This triplet is multiplied by a 3x3 matrix.

With this color transformation you can do your own color correction by changing the coefficients of the 3x3 matrix.

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{bmatrix} \times \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

Formula 1: Color transformation equation

Element	Description
R_{in}	Red component of the incoming RGB color pixel
G_{in}	Green component of the incoming RGB color pixel
B_{in}	Blue component of the incoming RGB color pixel
Gain00	Red contribution to the **red** pixel (multiplicative factor), ColorTransformationValueRR
Gain01	Green contribution to the **red** pixel (multiplicative factor), ColorTransformationValueRG
Gain02	Blue contribution to the **red** pixel (multiplicative factor), ColorTransformationValueRB
Gain10	Red contribution to the **green** pixel (multiplicative factor), ColorTransformationValueGR
Gain11	Green contribution to the **green** pixel (multiplicative factor), ColorTransformationValueGG
Gain12	Blue contribution to the **green** pixel (multiplicative factor), ColorTransformationValueGB
Gain20	Red contribution to the **blue** pixel (multiplicative factor), ColorTransformationValueBR

Table 30: Description for an RGB to RGB transformation

Element	Description
Gain21	Green contribution to the **blue** pixel (multiplicative factor), ColorTransformationValueBG
Gain22	Blue contribution to the **blue** pixel (multiplicative factor), ColorTransformationValueBB
R_{out}	Red resulting component of the pixel after the transformation
G_{out}	Green resulting component of the pixel after the transformation
B_{out}	Blue resulting component of the pixel after the transformation

Table 30: Description for an RGB to RGB transformation

Color correction is done in the same color space (RGB).

ColorTransformationControl

Camera control	Parameter	Description
ColorTransformationControl	ColorTransformationMode	*Off* When set to *Off*, no color correction is done.
		Manual When set to *Manual*, you can choose the coefficients of your color correction matrix, by setting each of the *ColorTransformationValueXY (with X=R or G or B and Y=R or G or B)* and entering the value.
		Temp6500K When set to *Temp6500K*, the colors will be optimized for a surrounding of color temperature 6500 K.
	ColorTransformationValueBB	Read and Write Values for the color transformation
	ColorTransformationValueBG	
	ColorTransformationValueBR	
	ColorTransformationValueGB	
	ColorTransformationValueGG	
	ColorTransformationValueGR	
	ColorTransformationValueRB	
	ColorTransformationValueRG	
	ColorTransformationValueRR	

Table 31: Camera control: **ColorTransformationMode**

DSP

The automatic exposure, gain, and white balance features can be configured to respond only to a subregion within the image scene. Use this feature to choose a subregion that will *meter* the rest of the image. This feature works like the region metering on a photographic camera. It is the portion of the image used to make the measurements required to adjust the automatic exposure and white balance features.

Camera control	Parameter	Description
DSP	DSPSubregionBottom	[Uint32] R/W Defines the bottom of the region in pixels. Defaults to a huge number much larger than the maximum number of sensor rows.
	DSPSubregionLeft	[Uint32] R/W Defines the position of left edge of the DSP subregion. Measured in pixels from the left edge. Defaults to zero.
	DSPSubregionRight	[Uint32] R/W Defines the position of the right edge of the DSP subregion as measured from the left side of the image. Defaults to a huge number much larger than the maximum number of sensor columns.
	DSPSubregionTop	[Uint32] R/W Defines the top edge of the DSP subregion defined as the number of pixels from the top edge of the full image. Defaults to zero.

Table 32: Camera control: **DSP**

EdgeFilter (only color models)

Controls relating to the sharpness of images (picture edge sharpness).

Camera control	Parameter	Description
EdgeFilter	Smooth2	Least sharp
	Smooth1	Less sharp
	Off	Default: no sharpness applied in either direction
	Sharpen1	Some sharp
	Sharpen2	Most sharp

Table 33: Camera control: **EdgeFilter**

Gamma

Camera control	Parameter	Description
Gamma		If Gamma is enabled, LUT position 1 contains Gamma values. The original LUT values will be stored temporarily. If Gamma is disabled, LUT position 1 contains optional user defined LUT values.
		1.000 for Gamma OFF (no Gamma correction) (Manta: Gamma=0.45 or 0.5 or 0.7) For Gamma ON: write value (e.g. for Manta: Gamma=0.45 or 0.5 or 0.7) In general: If you write an arbitrary value between 0 and 1, the next possible Gamma value (depending on AVT GigE camera) is chosen automatically. Examples for Manta camera: Write 0, the next possible value 0.45 is chosen. Write 0.51, the next possible value 0.7 is chosen. If Gamma is ON, and you read out LUT1: you only get stored LUT values but not Gamma values. In general: Gamma values can‘t be read out.

Table 34: Camera control: **Gamma**

Exposure

Minimum exposure time for the different Manta models can be found below. There is no exposure time offset.

Camera model	Exposure time offset
Manta G-032	0 µs
Manta G-033	0 µs
Manta G-046	0 µs
Manta G-125	0 µs
Manta G-145	0 µs
Manta G-145-30fps	0 µs
Manta G-146	0 µs
Manta G-201	0 µs
Manta G-201-30fps	0 µs
Manta G-504	0 µs

Table 35: Camera-specific exposure time offset

Camera model	Minimum exposure time
Manta G-032	26 µs
Manta G-033	26 µs
Manta G-046	26 µs
Manta G-125	21 µs
Manta G-145	38 µs
Manta G-145-30fps	37 µs
Manta G-146	31 µs
Manta G-201	51 µs
Manta G-201-30fps	10 µs
Manta G-504	38 µs

Table 36: Camera-specific minimum exposure time

Jitter at start of exposure

The following chapter discusses the latency time which exists for all Manta CCD models when either a hardware or software trigger is generated, until the actual image exposure starts.

Owing to the well-known fact that an **Interline Transfer CCD** sensor has both a light sensitive area and a separate storage area, it is common to interleave image exposure of a new frame and output that of the previous one. It makes continuous image flow possible, even with an external trigger.

The uncertain time delay before the start of exposure depends on the state of the sensor. A distinction is made as follows:

FVal is active → the sensor is reading out, the camera is busy

In this case the camera must not change horizontal timing so that the trigger event is synchronized with the current horizontal clock. This introduces a maximum uncertainty which is equivalent to the line time. The line time depends on the sensor used and therefore can vary from model to model.

FVal is inactive → the sensor is ready, the camera is idle

In this case the camera can resynchronize the horizontal clock to the new trigger event, leaving only a very short uncertainty time of the master clock period.

Model	Exposure start jitter (while FVal)	Exposure start jitter (while camera idle)
Manta G-032	± 24.3 µs	± 3.0 µs
Manta G-033	± 22.5 µs	± 1.5 µs
Manta G-046	± 25.1 µs	± 1.8 µs
Manta G-125	± 33.2 µs	± 5.0 µs
Manta G-145	± 57.8 µs	± 5.9 µs
Manta G-145-30fps	± 30.7 µs	± 4.7 µs
Manta G-146	± 53.5 µs	± 10.5 µs
Manta G-201	± 54.8 µs	± 7.1 µs
Manta G-201-30fps	± 26.9 µs	± 5.7 µs
Manta G-504	± 52.6 µs	± 10.3 µs

Table 37: Jitter at exposure start (no binning, no sub-sampling)

Note

- Jitter at the beginning of an exposure has no effect on the length of exposure, i.e. it is always constant.

Auto

This group of controls relates to the camera auto-exposure function. The camera controls, such as *ExposureAutoAdjustTol*, *ExposureAutoAlg* etc. can be adjusted to match imaging conditions.

Note

The camera must be acquiring images in order for the auto exposure algorithm to update.

Camera control	Parameter	Description
Auto	ExposureAutoAdjustTol	[Uint32] R/W In percent, from 0 to 50. A threshold. **Usage:** Sets a range in variation from *ExposureAutoTarget* in which the auto exposure algorithm will not respond. Use to limit exposure setting changes to only larger variations in scene lighting.

Table 38: Camera control: **ExposureAuto**

Camera control	Parameter	Description
Auto	ExposureAutoAlg	[Enum] R/W Use these algorithms to calculate auto exposure:
		Mean The arithmetic mean of the histogram of the current image is compared to *ExposureAutoTarget*, and the next image adjusted in exposure time to meet this target. Bright areas are allowed to saturate.
		FitRange The histogram of the current image is measured, and the exposure time of the next image is adjusted so bright areas are not saturated. In most cases, the *Mean* setting is preferable.
	ExposureAutoMax	[Uint32] R/W In microseconds. This sets the upper bound to the exposure setting in auto exposure mode. This is useful in situations where frame rate is important. This value would normally be set to something less than (as a rough estimate) $1x10^6$/(desired frame rate).
	ExposureAutoMin	[Uint32] R/W In microseconds. This sets the lower bound to the exposure setting in auto exposure mode.
	ExposureAutoOutliers	[Uint32] R/W Each unit represents 0.01%. When value is 1000, this equals 10%. The percentage defines the total pixels from top of the distribution that are ignored by the auto exposure algorithm.

Table 38: Camera control: **ExposureAuto**

Camera control	Parameter	Description
Auto	ExposureAutoRate	[Uint32] R/W In percent. 100% is auto-exposure adjustments running at full speed, 50% is half speed. Determines the rate at which the auto exposure function changes the exposure setting.
	ExposureAutoTarget	[Uint32] R/W In percent. Controls the general lightness or darkness of the auto exposure feature; specifically the target mean histogram level of the image: 0 being black 100 being white

Table 38: Camera control: **ExposureAuto**

ExposureMode

Exposure is known in the FireWire universe as shutter.

Camera control	Parameter	Description
ExposureMode		[Enum] R/W
		Manual The camera exposure time is fixed by *ExposureValue* parameter.
		AutoOnce [Command] When set to *AutoOnce*, the exposure will be set once according to the scene illumination: After that the ExposureMode is set to *Manual*. The *AutoOnce* exposure function operates according to the *Auto* and *DSP* controls.
		Auto When set to *Auto*, the exposure time will vary continuously according to the scene illumination. The *Auto* exposure function operates according to the *Auto* and *DSP* controls.
		External When *ExposureMode* is set to *External* the exposure time will be controlled by an external signal appearing on *SyncIn1* or *SyncIn2*. In order for this feature to work, the parameter *FrameStartTriggerMode* must be set to *SyncIn1* or *SyncIn2*.

Table 39: Camera control: **ExposureMode**

ExposureValue

Camera control	Parameter	Description
ExposureValue		[Uint32] R/W In microseconds. This is the sensor integration time. Examples: 15000 corresponds to 15 ms integration time. 1000 corresponds to 1 ms, etc.

Table 40: Camera control: **ExposureValue**

Gain

Auto

This group of controls relates to the camera auto gain function.

The auto gain function of AVT Manta cameras can be controlled by various parameters. The parameters, such as *GainAutoAdjustDelay*, *GainAutoAdjustTol*, etc. can be adjusted to match imaging conditions.

Application For example, in traffic imaging, the *GainAutoAdjustDelay* parameter might be increased so that momentary changes in scene lighting, such as on-coming headlights, do not affect the gain settings.

Note

The camera must be acquiring images in order for the auto gain algorithm to update.

Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

Camera control	Parameter	Description
GainAuto	GainAutoAdjustTol	[Uint32] R/W In percent, from 0 to 50. Sets a tolerance in variation from GainAutoTarget in which the auto exposure algorithm will not respond. **Usage:** Use this parameter to limit auto gain changes to only larger variations in scene lighting.
	GainAutoMax	[Uint32] R/W In dB. Sets the **upper** bound to the gain setting in auto gain mode.
	GainAutoMin	[Uint32] R/W In dB. Sets the **lower** bound to the gain setting in auto gain mode. **Usage:** Normally this number would be set to zero.
	GainAutoOutliers	[Uint32] R/W The *GainAutoOutliers* is the percentage of the image pixels that do not have to fit into the proper brightness range. **Usage:** Use this parameter to manipulate how the auto gain function works.
	GainAutoRate	[Uint32] R/W The parameter determines the rate at which the auto gain function changes. The value is a percentage of the maximum rate.
	GainAutoTarget	[Uint32] R/W This parameter controls the general lightness or darkness of the auto gain feature. The value is a percentage of maximum brightness.

Table 41: Camera control: **GainAuto**

GainMode

Camera control	Parameter	Description
GainMode		[Enum] R/W
	Manual	When set to *Manual*, the camera gain is fixed by GainValue parameter.
	AutoOnce	[Command] When set to *AutoOnce*, the gain will be set once according to the scene illumination. After that the ExposureMode is set to *Manual*. The *AutoOnce* gain function operates according to the *Auto* and *DSP* controls.
	Auto	When set to *Auto*, the gain will vary continuously according to the scene illumination. The *Auto* gain function operates according to the *Auto* and *DSP* controls.

Table 42: Camera control: **GainMode**

GainValue

Camera control	Parameter	Description
GainValue		[Uint32] R/W In dB. $G_{dB} = 20 \log_{10}(V_{out}/V_{in})$. This is the gain setting applied to the sensor. **Usage:** For best image quality, the gain setting should be set to zero. However, in low-light situations, it may be necessary to increase the gain setting.

Table 43: Camera control: **GainValue**

Hue (only color models)

Camera control	Parameter	Description
Hue		In degrees (-40° ... +40°) All values are rounded off to integers. Changes the color of objects without altering the white balance from the nominal perception. **Usage:** Use to manipulate the color appearance after having carried out the white balance. Does not show any effect in Raw formats, because color processing is switched off in all Raw formats.

Table 44: Camera control: **Hue**

Iris

(Firmware 1.44 or greater)

All video-type **auto-iris** lenses have a default reference voltage. When a voltage larger than this reference voltage is applied to the lens, the iris closes. When a voltage is applied less than this reference voltage, the iris opens.

The auto iris algorithm calculates the appropriate voltage, *IrisVideoLevel*, to apply to the lens, based on the information of the current image. The camera must be acquiring images in order for the auto algorithm to update. Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

The auto-iris feature can be configured to respond only to a subregion within the image scene.

Camera control	Parameter	Description
Iris	IrisAutoTarget	[Uint32] R/W In percent. Controls the general lightness or darkness of the auto iris feature; specifically the target mean histogram level of the image, 0 being black, 100 being white.
	IrisMode	[Enum] R/W Sets the auto-iris mode. *Disabled* Turn off the video auto-iris function. *Video* Turn on the video auto-iris function. *VideoOpen* Fully open the iris. *VideoClosed* Full close the iris.
	IrisVideoLevel	[Uint32] R/W In 10 mV units. This attribute reports the strength of the video signal coming from the camera.
	IrisVideoLevelMax	[Uint32] R/W In 10 mV units. Limits the **maximum** driving voltage for closing the lens iris. Typically this will be 150, however it may vary dependent on the lens reference voltage.
	IrisVideoLevelMin	[Uint32] R/W In 10 mV units. Limits the **minimum** driving voltage for opening the lens iris. Typically this will be 0.

Table 45: Camera control: **Iris (video auto-iris lens)**

LUTControl

(Firmware 1.44 or greater)

With this control, the look-up table (LUT) related features can be controlled.

In principle, there are two methods to work with LUTs:

- The LUT feature allows to write index/values into the camera (approximately 5 seconds for upload).
- Alternatively the feature can work with direct memory access to LUT address via block upload: 2 LUT values are packed as 16-bit word in 32-bit register. The maximum size for block-wise upload is limited by the PvAPI to 512 Byte. So you need 16 blocks to upload one complete LUT. This block mode is the fastest method (less than 1 second for upload).

Working with direct memory access: block upload (Example scenario)

Perform the following steps for each of LUT1/LUT2/LUT3:

1. Decide which LUT to use and choose LUT1 or LUT2 or LUT3. All further steps are applied to the chosen LUT.
2. Get value range for LUTIndex (0 ... 4095 that are 4096 entries).
3. Get value range for LUTValue, that are values for the LUT entries (0 ... 4095 that are 4096 values)
4. Check LUTBitDepthIn. Manta: 12 bit (2^{12}=4096, the value of LUTIndex)
5. Check LUTBitDepthOut. Manta: 12 bit (2^{12}=4096, the value of LUTValue)
6. Check LUTAddress: memory address in camera
7. Check LUTSizeBytes. That is the look-up table size in Bytes (8 KBytes)
8. Create a LUT memory array in PC (according to LUTIndex and LUTValue)
9. Fill memory array with LUT values (change endianess of every 2 Bytes in LUT entry)
10. Write memory array from PC into camera (blockwise 512 Byte until all data are written in camera) beginning at LUTAddress
11. Execute LUTSave command.

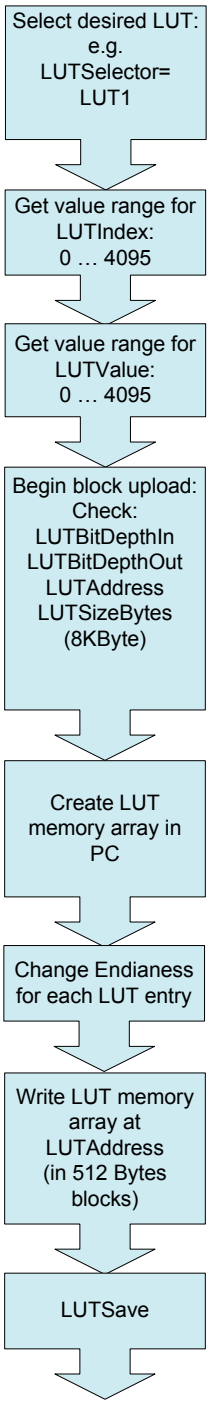


Figure 23: Loading an LUT

LUTControl

LUTControl is a category that includes the LUT control feature.

The use of one LUT allows any function (in the form Output = F(Input)) to be stored in the camera's RAM and to be applied on the individual pixels of an image at run-time.

The address lines of the RAM are connected to the incoming digital data, these in turn point to the values of functions which are calculated offline, e.g. with a spreadsheet program.

This function needs to be loaded into the camera's RAM before use.

Note **Color cameras only:**

If the color information of an image was destroyed by using the binning functionality, than the LUTControl feature with single color planes will not work correctly.

LUTSelector

Camera control	Parameter	Description
LUTSelector		[Enum] Possible values: *LUT1*, *LUT2*, *LUT3* Select which LUTs to control. These LUTs are device specific. If Gamma is ON: LUT1 contains gamma values.

Table 46: Camera control: **LUTSelector**

LUTMode

Here you can select (depending on LUTSelector) on which pixels the LUT will be applied.

Camera control	Parameter	Description
LUTMode		[Enum] R/W
		Red When set to *Red*, LUT is applied on red pixels only. Available for color cameras only.
		Green When set to *Green*, LUT is applied on green pixels only. Available for color cameras only.
		Blue When set to *Blue*, LUT is applied on blue pixels only. Available for color cameras only.
		Luminance When set to *Luminance*, LUT is applied on all pixels. Available for b/w and color cameras.

Table 47: Camera control: **LUTMode**

LUTEnable

Camera control	Parameter	Description
LUTEnable		[Boolean] *True/false* When set to *true*: activates the selected LUT. When set to *false*: deactivates the selected LUT.

Table 48: Camera control: **LUTEnable**

LUTIndex

Camera control	Parameter	Description
LUTIndex		[Integer] Controls the index (offset) of the coefficient to access in the selected LUT.

Table 49: Camera control: **LUTIndex**

LUTValue

Camera control	Parameter	Description
LUTValue		[Integer] Returns the value at entry LUTIndex of the LUT selected by LUTSelector.

Table 50: Camera control: **LUTValue**

LUTLoad

Camera control	Parameter	Description
LUTLoad		[Command] Loads LUT from internal memory of the camera.

Table 51: Camera control: **LUTLoad**

LUTSave

Camera control	Parameter	Description
LUTSave		[Command] Saves LUT into internal memory of the camera. **Note** With ConfigFile control (ConfigFileSave command) you can't save the contents of the LUT.

Table 52: Camera control: **LUTSave**

LUTInfo

This control provides information depending on LUTSelector.

Camera control	Parameter	Description
LUTAddress		[Integer] Indicates location of memory, when LUT is loaded.
LUTSizeBytes		[Integer] Number of bytes from LUT Address memory.
LUTBitDepthIn		[Integer] Number of bit depth (12 bit) for signal coming into LUT.
LUTBitDepthOut		[Integer] Number of bit depth (12 bit) of signal going out of LUT.

Table 53: Camera control: **LUTInfo**

Offset

Offset is known in the FireWire standard as brightness (aka black level).

Camera control	Parameter	Description
Offset		0 ... 255 Note: Setting the gain does not change the offset (brightness, black level).

Table 54: Camera control: **Brightness**

Saturation (only color models)

Camera control	Parameter	Description
Saturation		Units: 0 ... 2.0 Changes the intensity of the colors. Examples: 0: no color saturation (monochrome image) 1: standard 2: max. color saturation Does not show any effect using Raw formats, because color processing is switched off in all Raw formats.

Table 55: Camera control: **Saturation**

WhiteBalance

WhitebalMode

There are three types of white balance:

- **Manual**
- **Auto**
- **AutoOnce**

White balance is applied so that non-colored image parts are displayed non-colored.

Auto The **WhitebalMode Auto** continuously optimizes the color characteristics of the image.

For the white balance algorithm the whole image or a subset of it is used.

AutoOnce The camera automatically generates frames, based on the current settings of Gain, Offset, Exposure, etc.

For the white balance algorithm the whole image or a subset (area defined by DSPSubregion) of it is used.

WhitebalMode AutoOnce acts like following: A histogram is taken by the algorithm and color balance is adjusted until each channel is equal. In detail: the algorithm checks the difference between channels and goes back to manual mode when the channels are within *WhitebalAutoAdjustTol* percent.

This feature uses the assumption that the image is gray/white on average.

The following flow diagram illustrates **WhitebalMode AutoOnce.**

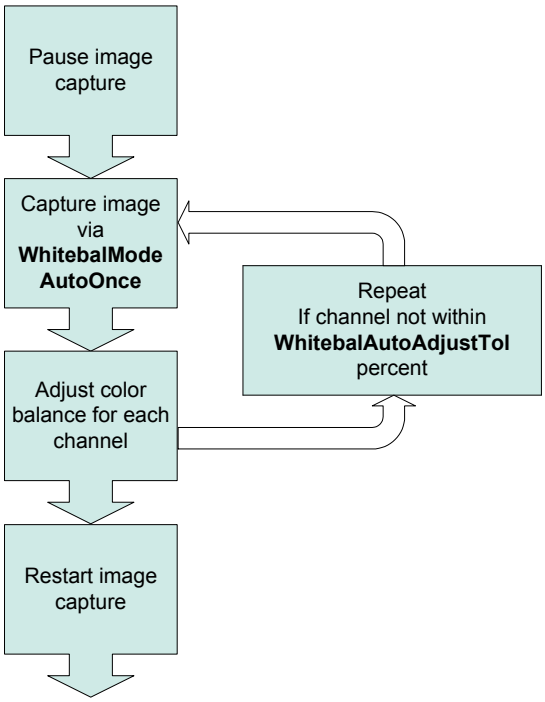


Figure 24: **WhitebalMode AutoOnce**

Auto

Auto algorithms use information from the camera's current image and apply the following settings to the next image. I.e. the camera must be acquiring images in order for the auto algorithm to update. Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

Camera control	Parameter	Description
Auto	WhitebalAutoAdjustTol	[Uint32] R/W A threshold. This parameter sets a range of scene color changes in which the auto white balance will not respond. **Usage:** Use to limit white balance setting changes to only larger variations in scene color.
	WhitebalAutoRate	[Uint32] R/W In percent. Determines how fast the auto white balance updates. **Usage:** Use to slow the rate of color balance change so that only longer period fluctuations affect color.

Table 56: Camera control: **WhiteBalanceAuto**

Camera control	Parameter	Description
WhitebalMode		[Enum] R/W
	Manual	Auto white balance is off. **Usage:** White balance can be adjusted directly by changing the *WhitebalValueRed* and *WhitebalValueBlue* parameters.
	Auto	White balance will continuously adjust according to the current scene. The Auto function operates according to the *Auto* and *DSP* controls
	AutoOnce	A command (of type Enumeration in AVT Universal Package). A single iteration of the auto white balance algorithm is run, and then the camera reverts to *Manual WhitebalMode*. The AutoOnce function operates according to the *Auto* and *DSP* controls

Table 57: Camera control: **WhitebalMode**

Manual The manual settings have the following range:

Type	Range	Range in dB	Increment length
Manta color cameras	80 ... 300 (80 equals 0 dB)	+ 6.85 dB	~0.031 dB/step

Table 58: Manual gain range of the various Manta types

WhitebalValueRed

Camera control	Parameter	Description
WhitebalValueRed		[Uint32] R/W Red gain expressed as a percentage of the camera default setting.

Table 59: Camera control: **WhitebalValueRed**

WhitebalValueBlue

Camera control	Parameter	Description
WhitebalValueBlue		[Uint32] R/W Blue gain expressed as a percentage of the camera default setting.

Table 60: Camera control: **WhitebalValueBlue**

EventControl

(Firmware 1.44 or greater)

The following table lists all the events supported by the camera:

Camera control	Parameter	ID / Description
EventID	EventAcquisitionStart	[Uint32] R/C 40000
	EventAcquisitionEnd	[Uint32] R/C 40001
	EventFrameTrigger	[Uint32] R/C 40002
	EventExposureEnd	[Uint32] R/C 40003
	EventAcquisitionRecordTrigger	[Uint32] R/C 40004
	EventSyncIn1Rise	[Uint32] R/C 40010
	EventSyncIn1Fall	[Uint32] R/C 40011
	EventSyncIn2Rise	[Uint32] R/C 40012
	EventSyncIn2Fall	[Uint32] R/C 40013
	EventSyncIn3Rise	[Uint32] R/C 40014
	EventSyncIn3Fall	[Uint32] R/C 40015
	EventSyncIn4Rise	[Uint32] R/C 40016
	EventSyncIn4Fall	[Uint32] R/C 40017
	EventOverflow	[Uint32] R/C 65534
	EventError	[Uint32] R/C 65535 Always on. Cannot be turned off with EventSelector or EventsEnable1. Event should never occur, only returning in case of firmware failure requiring camera repair.

Table 61: Camera control: **EventID**

Camera control	Parameter	ID / Description
EventNotification		[Enum] R/W Turns the selected event notification *On* or *Off*.
EventSelector		[Enum] R/W Select a specific event to be enabled or disabled using EventNotification *AcquisitionStart* *AcquisitionEnd* *FrameTrigger* *ExposureEnd* *AcquisitionRecordTrigger* *SyncIn1Rise* *SyncIn1Fall* *SyncIn2Rise* *SyncIn2Fall* *SyncIn3Rise* *SyncIn3Fall* *SyncIn4Rise* *SyncIn4Fall*
EventsEnable1		[Uint32] R/W Bitmask of events currently enabled. Bit 1 is *EventAcquisitionStart* Bit 2 is *EventAcquisitionEnd* Bit 3 is *FrameTrigger*, and so on. This is an alternative to setting each of the event individually using the *EventNotification* and *EventSelector* method.

Table 61: Camera control: **EventID**

ConfigFile (user sets)

AVT Manta cameras are capable of storing a number of user-specified configurations within the camera's non-volatile memory. These saved configurations can be used to define the power-up settings of the camera or to quickly switch between a number of predefined settings.

Note

With this control you can save the state of the LUT, but the contents of the LUT is **not** saved.

To save the content of a LUT, use Controls/LUTControl/LUTSave.

Camera control	Parameter	Description
ConfigFile	ConfigFileIndex	[Enum] R/W This is the index number corresponding to the configuration set that you are currently working with. Possible settings: Factory, 1, 2, 3, 4, 5
	ConfigFileLoad	[Command] **Usage:** Use to load the configuration corresponding to *ConfigFileIndex*.
	ConfigFilePowerUp	[Enum] R/W This control sets the memory index number that determines which saved configuration will load when the camera powers up. Possible settings: Factory, 1, 2, 3, 4, 5
	ConfigFileSave	[Command] **Usage:** Use to save the current camera settings into the non-volatile memory location currently indicated by *ConfigFileIndex*. The *Factory* setting cannot be overwritten.

Table 62: Camera control: **ConfigFile**

GigE

BandwidthCtrlMode

Camera control	Parameter	Description
BandwidthCtrlMode		[Enum] R/W Select the desired mode of bandwidth control.
		StreamBytesPerSecond The **default** mode of bandwidth control. This is the recommended method of regulating the output data rate of the camera. **Usage:** Use for managing bandwidth allocation between multiple cameras, and for managing the data flow into the host computer. See the *StreamBytesPerSecond* control for more information: Chapter StreamBytesPerSecond on page 91.
		SCPD Stream channel packet delay expressed in timestamp counter units. This mode is **not** recommended.
		Both Implements a combination of control modes. This mode is **not** recommended.

Table 63: Camera control: **BandwidthCtrlMode**

ChunkModeActive

(Firmware 1.44 or greater)

Camera control	Parameter	Description
ChunkModeActive		[Boolean] R/W Possible values: *True*, *False* Enables PvAPI tPvFrame→AncillaryBuffer to receive associated chunk mode data from the camera. See the **PvAPI Programmers' Reference Manual** for more information.

Table 64: Camera control: **ChunkModeActive**

NonImagePayloadSize

(Firmware 1.44 or greater)

Camera control	Parameter	Description
NonImagePayloadSize		[Uint32] R/W Size of chunk mode data, in bytes.

Table 65: Camera control: **NonImagePayloadSize**

PayloadSize

Camera control	Parameter	Description
PayloadSize		[Uint32] R/W Total size of payload in bytes. Payload = TotalBytesPerFrame + NonImagePayloadSize + 8

Table 66: Camera control: **PayloadSize**

StreamFrameRateConstrain

Camera control	Parameter	Description
StreamFrameRateConstrain		[Boolean] R/W Possible values: *True*, *False* When *True*, camera automatically limits frame rate to bandwidth, determined by StreamBytesPerSecond, to prevent camera buffer overflows and dropped frames. If *False*, frame rate is not limited to bandwidth – only sensor readout time. Latter case is useful for *AcquisitionMode* = *Recorder* or *StreamHoldEnable* modes.

Table 67: Camera control: **StreamFrameRateConstrain**

Ethernet

Camera control		Description
Ethernet	DeviceEthAddress	[String] R/C The physical MAC address of the camera
	HostEthAddress	[String] R/C The physical MAC address of the host network card

Table 68: Camera control: **Ethernet**

IP

Camera control	Parameter	Description
IP	DeviceIPAddress	[String] R/C The current IP address of the camera
	HostIPAddress	[String] R/C The current IP address of the host network interface

Table 69: Camera control: **IP**

GvcpRetries

AVT Manta cameras have a sophisticated real-time resend mechanism that ensures a high degree of data integrity.

Camera control	Parameter	Description
GvcpRetries		Controls the maximum number of resend requests that the host will attempt when trying to recover a lost packet.

Table 70: Camera control: **GvcpRetries**

Gvsp

Definition Gvsp = GigE Vision Streaming Protocol

Camera control	Parameter	Description
Gvsp	GvspLookbackWindow	[Uint32] R/W Size of the look back window, in packets, when determining if a stream packet is missing.
	GvspResentPercent	[Float32] R/W Maximum of missing stream packets that will be requested from the camera if they are detected missing.
	GvspRetries	[Uint32] R/W The maximum number of resend requests that the host will attempt when trying to recover a lost stream packet.
	GvspSocketBufferCount	[Enum] R/W Number of buffers to be used by the network socket. Only applicable when not using the Filter Driver. Possible values: 256, 512, 1024, 2048, 4096, 8192
	GvspTimeout	[Uint32] R/W End of stream timeout, in milliseconds.

Table 71: Camera control: **Gvsp**

HeartbeatInterval

Camera control	Parameter	Description
HeartbeatInterval		[Uint32] R/W In milliseconds. **Definition:** The interval at which the API sends a heartbeat command to the camera. **Usage:** Normally this parameter does not require adjustment.

Table 72: Camera control: **HeartbeatInterval**

HeartbeatTimeout

Camera control	Parameter	Description
HeartbeatTimeout		[Uint32] R/W In milliseconds. **Definition:** The maximum amount of time the camera will wait for a heartbeat command before timing out. **Usage:** Normally this parameter does not require adjustment. **Note** This value may need to be increased when using breakpoints in your API code. Breakpoints stall the API from sending heartbeat commands, which may cause the camera to time out.

Table 73: Camera control: **HeartbeatTimeout**

Multicast

Multicast mode allows the camera to send image data to all hosts on the same subnet as the camera.

The host computer (or SampleViewer application instance) that first enables multicast mode is the master, and controls all camera parameters. All other hosts / instances are the monitors, and can view image data only.

Note Most GigE switches support a maximum PacketSize of 1500 in Multicast mode.

Camera control	Parameter	Description
MultiCast	MulticastEnable	[Enum] R/W Enables multicast mode. In multicast mode all computers on the same subnet as the camera can receive image data from the camera MulticastIPAddress. Values: On or Off Default: Off
	MulticastIPAddress	[String] R/W Sets the multicast IP address.

Table 74: Camera control: **Multicast**

PacketSize

Camera control	Parameter	Description
PacketSize		[Uint32] R/W In Byte. This parameter determines the Ethernet packet size. **Usage:** Generally speaking this number should be set to as large as the network card will allow. If this number is reduced, then CPU loading will increase. These large packet sizes are called **Jumbo Packets/Frames** in Ethernet terminology. If your Gigabit Ethernet network card does not support **Jumbo Packets/Frames** of at least 9 kByte, then you will need to reduce PacketSize parameter of the camera to match the maximum Jumbo packet size supported by your Gigabit Ethernet interface. If you don't know what this value is, you may need to decrease PacketSize to 1500 which all Gigabit Ethernet cards support. **Note** If you are seeing all *black images*, or all frames reported as *StatFramesDropped* and zero images reported as *StatFramesCompleted*, you will likely need to decrease this parameter.

Table 75: Camera control: **PacketSize**

StreamBytesPerSecond

Camera control	Parameter	Description
StreamBytesPerSecond		[Uint32] R/W **Usage:** Use to moderate the data rate of the camera. This is particularly useful for slowing the camera down so that it can operate over slower links such as Fast Ethernet (100 Mbit/s), or wireless networks. It is also an important control for multi-camera situations. When multiple cameras are connected to a single Gigabit Ethernet port (usually through a switch), *StreamBytesPerSecond* for each camera needs to be set to a value so that the sum of each cameras *StreamBytesPerSecond* parameter does not exceed the data rate of the GigE port. Setting this parameter in this way will ensure that multiple camera situations work without data loss. 115,000,000 is the typical data maximum data rate for a GigE port. To calculate the required minimum *StreamByetsPerSecond* setting for a camera in any image mode, use the following formula: Height x Width x FrameRate x Bytes per Pixel (see Chapter ImageFormat on page 48) **Note** If you are seeing occasional *black images*, or occasional frames/ packets reported as *StatFramesDropped/ StatPacketsDropped* you will likely need to decrease this parameter.

Table 76: Camera control: **StreamBytesPerSecond**

StreamHold

This is an interesting feature for controlling when the camera sends data to the host computer. Normally the camera sends data to the host computer immediately after completion of exposure. Enabling *StreamHold* delays the transmission of data, storing it in on-camera memory, until StreamHold is disabled.

This feature can be useful to prevent GigE network flooding in situations where a large number of cameras connected to a single host computer are capturing a single event. Using the *StreamHold* function, each camera will hold the event image data until the host computer disables *StreamHold* for each camera in turn.

Camera control	Parameter	Description
StreamHold	StreamHoldCapacity	[Uint32] R/V This read-only value represents the total number of image frames that can be stored in the camera memory. This value is different for each camera depending on the camera internal memory size and the *TotalBytesPerFrame*.
	StreamHoldEnable	[Enum] R/W Possible values: *On*, *Off* This enables streamhold functionality. When disabled, the image data will be released to the host computer.

Table 77: Camera control: **StreamHold**

Timestamp

AVT Manta cameras have a very accurate timestamp function for timestamping images.

Camera control	Parameter	Description
Timestamp	TimeStampFrequency	[Uint32] R/C In Hz. All images returned from the camera are marked with a timestamp. *TimeStampFrequency* is the time base for the timestamp function. **Usage:** The image timestamp can be useful for determining whether images are missing from a sequence due to missing trigger events.
	TimeStampReset	[Command] Reset the camera's time stamp to 0.
	TimeStampValueHi	[Uint32] R/V Time stamp, upper 32-bit.
	TimeStampValueLo	[Uint32] R/V Time stamp, lower 32-bit.
	TimeStampValueLatch	[Command] Latch the value of the timestamp on the camera. Both TimeStampValueHi and TimeStampValueLo are updated with the value read from the camera.

Table 78: Camera control: **Timestamp**

IO

The control and readout of all camera inputs and outputs. The number of inputs and outputs will depend on your camera model.

Strobe

Definition Strobe is an internal signal generator for on-camera clocking functions.

Valid when any of the SyncOut modes are set to Strobe1. Strobe allows the added functionality of duration and delay, useful when trying to sync a camera exposure to an external strobe.

1 - Settings specific to Strobe1

Camera control	Parameter	Description
1	Strobe1ControlledDuration	[Enum] R/W Possible Values: *On*, *Off* Enables/disables control over length and start delay of strobe signal
	Strobe1Delay	[Uint32] R/W In microseconds. Delay of start of strobe signal
	Strobe1Duration	[Uint32] R/W In microseconds. Duration of strobe signal.

Table 79: Camera control: **Strobe1**

Camera control	Parameter	Description
1	Strobe1Mode	[Enum] R/W Associates the start of strobe signal with one of the following image capture events:
		AcquisitionTriggerReady Becomes active once the camera has been recognized by the host PC and is ready to start acquisition.
		FrameTriggerReady Becomes active when the camera is in a state that will accept the next frame trigger.
		FrameTrigger This is the logic trigger signal inside of the camera. It is initiated by an external trigger or software trigger.
		Exposing Becomes active at the start of exposure.
		FrameReadout Becomes active at the start of frame readout.
		Acquiring Becomes active at the start of acquisition.
		SyncIn1 Active when there is an external trigger at syncIn1.
		SyncIn2 Active when there is an external trigger at syncIn2.

Table 79: Camera control: **Strobe1**

Note

For detailed information see the camera waveform diagrams in Chapter Trigger concept for advanced users on page 43ff.

SyncIn1

Use SyncIn1 as a general purpose input (GPI).

Camera control	Parameter	Description
SyncIn1	SyncInGlitchFilter	[Uint32] R/W In nanoseconds. Ignores glitches on the SyncIn input line with pulse duration less than set value. **Note** Setting this value increases latency of FrameTrigger by same amount.

Table 80: Camera control: **SyncIn1**

SyncIn2

Analogous to SyncIn1

SyncLevels

Camera control	Parameter	Description
SyncInLevels		[Uint32] R/V Read only. This is a 4-bit register where each bit corresponds to a specific SyncIn input. **Example:** When this value returns *2* for instance (0010) SyncIn2 is high and all other Sync input signals are low.

Table 81: Camera control: **SyncInLevels**

SyncOut

Controls for configuring the output trigger used for synchronization with other cameras/devices or the camera outputs.

Usage: Use for synchronization with other cameras/devices or general purpose outputs.

SyncOut1

Camera control	Parameter	Description
Syncout1	SyncOut1Invert	[Enum] R/W Possible values: *On*, *Off* When enabled, reverses the polarity of the signal output by SyncOut1.
	SyncOut1Mode	[Enum] R/W Determines the type of output defined by SyncOut1:
		GPO Configured to be a general purpose output, control of which is assigned to *SyncOutGpoLevels*

Table 82: Camera control: **SyncOut1**

Camera control	Parameter	Description
SyncOut1	SyncOut1Mode	*AcquisitionTriggerReady* Active once the camera has been recognized by the host PC and is ready to start acquisition.
		FrameTriggerReady Active when the camera is in a state that will accept the next frame trigger.
		FrameTrigger Active when an image has been initiated to start. This is a logic trigger internal to the camera, which is initiated by an external trigger or software trigger event.
		Exposing Active for the duration of sensor exposure.
		FrameReadout Active at during frame readout, i.e. the transferring of image data from the CCD to camera memory.
		Imaging Exposing or frame readout. Active when the camera is exposing or reading out frame data
		Acquiring Active when acquisition start has been initiated
		SyncIn1 Active when there is an external trigger at SyncIn1
		SyncIn2 Active when there is an external trigger at SyncIn2
		Strobe1 The output signal is controlled according to Strobe1 settings.

Table 82: Camera control: **SyncOut1**

Note For detailed information see the camera waveform diagrams in Chapter Trigger concept for advanced users on page 43ff.

SyncOut2

Note SyncOut2 Control is analogous to SyncOut1.

SyncOutGpoLevels

Camera control	Parameter	Description
SyncOutGpoLevels		GPO output levels. A bitfield. Bit 0 is sync-out 0 Bit 1 is sync-out 1, etc.

Table 83: Camera control: **SyncOutGpoLevels**

Stats

Camera control	Parameter	Description
Stats	StatDriverType	[Enum] R/V There are two main types of drivers currently available for use with AVT's cameras: Standard, and Filter. Using the Filter driver will reduce the load on the host CPU. If this value shows *Filter*, the filter driver is installed and is being used. If it reports *Standard*, then the filter driver is not installed, or is not activated.
	StatFilterVersion	[String] R/C Version of the filter driver being used.
	StatFrameRate	[Float32] R/V The current actual frame rate of the camera as received by the driver.
	StatFramesCompleted	[Uint32] R/V The number of frames captured since the start of imaging.
	StatFramesDropped	[Uint32] R/V The number of frames dropped during transmission since the start of imaging. **Note** If everything is configured correctly, this number should be zero. See *StreamBytesPerSecond*, *PacketSize*, and refer to the Host Computer Optimizations note.
	StatPacketsErroneous	[Uint32] R/V The number of improperly formed packets. If this number is non-zero, it suggests a possible cable or camera hardware failure.

Table 84: Camera control: **Stats**

Camera control	Parameter	Description
Stats	StatPacketsMissed	[Uint32] R/V The number of packets missed since the start of imaging. **Note** If everything is configured correctly, this number should remain zero, or at least very low compared to *StatPacketsReceived*. See *StreamBytesPerSecond*, *PacketSize*, and refer to Host Computer Optimizations note.
	StatPacketsReceived	[Uint32] R/V Indicates the number of packets received by the driver, this number should grow steadily during continuous acquisition.
	StatPacketsRequested	[Uint32] R/V When an expected packet is not received by the driver, it is recognized as missing and the driver requests the camera to resend it. The resend mechanism ensures very high data integrity. **Note** If everything is configured correctly, this number should remain zero, or at least very low compared to *StatPacketsReceived*. See *StreamBytesPerSecond*, *PacketSize*, and refer to Host Computer Optimizations note.
	StatPacketsResent	[Uint32] R/V The number of packets resent by the camera since the start of imaging.

Table 84: Camera control: **Stats**

Index

Numbers

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Y